

UNIVERZA V LJUBLJANI  
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Nejc Slejko

**Prototip preprostega sistema za  
upravljanje skladišča temelječega na  
mobilnih napravah Android in  
strežniku PHP**

DIPLOMSKO DELO  
VISOKOŠOLSKI STROKOVNI ŠTUDIJSKI PROGRAM PRVE  
STOPNJE RAČUNALNIŠTVO IN INFORMATIKA

MENTOR: doc. dr. Damjan Vavpotič

Ljubljana 2014



Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Tematika naloge:

V okviru diplomske naloge preučite možnosti za uporabo pametnih telefonov oz. drugih mobilnih pametnih naprav temelječih na operacijskem sistemu Android za potrebe skladiščenja. Pri tem predpostavite, da so vsi predmeti v skladišču označeni s črtnimi ali QR kodami, mobilna naprava pa naj omogoči njihovo branje. Razvijte prototip sistema, ki bo deloval na mobilni napravi, hkrati pa zasnujte tudi prototip strežnika, ki se bo povezal z mobilno napravo in bo omogočal centralizirano evidenco in nadzor skladišča. V okviru naloge tudi kratko predstavite uporabljene tehnologije pa tudi morebitne sorodne že obstoječe rešitve.



## IZJAVA O AVTORSTVU DIPLOMSKEGA DELA

Spodaj podpisani Nejc Slejko, z vpisno številko **63080175**, sem avtor diplomskega dela z naslovom:

*Prototip preprostega sistema za upravljanje skladišča temelječega na mobilnih napravah Android in strežniku PHP*

S svojim podpisom zagotavljam, da:

- sem diplomsko delo izdelal samostojno pod mentorstvom doc. dr. Damjana Vavpotiča,
- so elektronska oblika diplomskega dela, naslov (slov., angl.), povzetek (slov., angl.) ter ključne besede (slov., angl.) identični s tiskano obliko diplomskega dela,
- soglašam z javno objavo elektronske oblike diplomskega dela na svetovnem spletu preko univerzitetnega spletnega arhiva.

V Ljubljani, dne 19. november 2014

Podpis avtorja:



*Zahvaljujem se mentorju doc. dr. Damjanu Vavpotiču, za vso pomoč in nasvete pri izdelavi diplomskega dela.*

*Zahvalil bi se tudi svojim staršem in starim staršem za vso pomoč in podporo skozi vsa leta študija.*

*Zahvala gre tudi Jasmini Vrtovec za lektoriranje dela, ter vsem prijateljem in znancem, ki so me podpirali.*





# Kazalo

Slike

Povzetek

Abstract

<b>1</b>	<b>Uvod</b>	<b>1</b>
<b>2</b>	<b>Primerjava konkurenčnih aplikacij</b>	<b>3</b>
2.1	Pregled drugih aplikacij . . . . .	4
2.2	Povzetek pregleda . . . . .	7
<b>3</b>	<b>Uporabljene tehnologije</b>	<b>9</b>
3.1	Android . . . . .	9
3.2	REST . . . . .	9
3.3	BarcodeScanner . . . . .	10
<b>4</b>	<b>Analiza aplikacije</b>	<b>11</b>
4.1	Namen in uporaba aplikacije . . . . .	11
4.2	Delovanje na prenosni napravi Android . . . . .	12
4.3	Delovanje na strežniku - PHP in MySQL . . . . .	13
4.4	Prednost pred ostalimi sistemi . . . . .	14
<b>5</b>	<b>Arhitektura sistema</b>	<b>15</b>
5.1	Pogled na primere uporabe . . . . .	15

## KAZALO

5.2	Implementacijski pogled . . . . .	16
5.3	Procesni pogled . . . . .	17
5.4	Podatkovni pogled . . . . .	18
<b>6</b>	<b>Delovanje sistema</b>	<b>21</b>
6.1	Delovanje Android aplikacije . . . . .	21
6.2	Uporabljene knjižnice android . . . . .	26
6.3	Delovanje strežniških aplikacij . . . . .	28
<b>7</b>	<b>Zaključek</b>	<b>33</b>

# Slike

4.1	Primer dobro organiziranega skladišča. . . . .	12
4.2	Zaslonska maska aplikacije na Android napravi. . . . .	13
4.3	Povezava Android aplikacije s podatkovnim skladiščem. . . . .	14
5.1	Diagram primerov uporabe. . . . .	16
5.2	Komponentni diagram sistema v povezavi z vozlišči. . . . .	17
5.3	Porazdeljenost aktivnosti pri vnosu novega paketa. . . . .	18
5.4	ER diagram podatkovne baze. . . . .	19
6.1	Primer postopka skladiščenja. . . . .	23
6.2	Primer prikaza podatkov po poizvedbi. . . . .	24
6.3	Primer postopka prevzema dobavnice. . . . .	25
6.4	Vmesnik ZXing za branje črtnih kod. . . . .	26
6.5	Inicializacija funkcije za zagon kamere. . . . .	27
6.6	Prejemanje podatkov iz aktivnosti. . . . .	27
6.7	Pravilno pripravljena etiketa za označevanje paketa. . . . .	29
6.8	Zaslonska maska vmesnika za urejanje podatkov paketa. . . . .	30
6.9	Klic skripte za izpis informacije paketa. . . . .	31
6.10	JSON odgovor v primeru uspešnega klica. . . . .	31



# Povzetek

V diplomskem delu je predstavljena analiza konkurenčnih android aplikacij ter postopek načrtovanja in izdelave aplikacijskega sistema ter android aplikacije. Sistem bo deloval ločeno na napravi android in aplikacijskem strežniku, preko katerega bo prejemal podatke potrebne za delovanje. Na tržišču je veliko konkurenčnih aplikacij za skladiščenje. Naš doprinos je centralno skladiščenje podatkov in možnost uporabe večjega števila naprav naenkrat, kjer se podatki na vseh sinhronizirajo v realnem času.



# Abstract

This thesis presents an analysis of the competitive android applications, the process of planning and development of application system and android applications. The system will operate separately on android device and the application server, through which it will receive the information necessary for the operation. On the market there are many competing applications for storage management. Our contribution is the central storage of data and the ability to use multiple devices at once, where data is synchronized on all devices in real time.





# Poglavje 1

## Uvod

V večjih podjetjih, kjer morajo svoje izdelke skladiščiti v večjih količinah, se lahko hitro zgodi, da se določeni pri tem izgubijo ali založijo. Zato je prav, da se pakete skladišči po sistemu, ki nam lahko zagotovi, da lahko izdelke najdemo in do njih dostopamo kadar koli.

Naš sistem je zgrajen iz serverskega dela, ki ga sestavljata uporabniški vmesnik in avtomatika shranjevanja. Z uporabniškim vmesnikom imamo možnost določanja različnih parametrov in nastavitev, ki se uporabljajo pri sortiranju in skladiščenju paketov.

Na drugi strani imamo napravo za identifikacijo izdelkov in prikaz navodil. Pri tem je združenih več različnih tehnologij, v ospredju pa bo platforma Android. V današnjem času so te naprave zelo prilagodljive in lahko dostopne, dejansko lahko aplikacijo poganjamo tudi na mobilni napravi (pametni telefon). S tem razširimo nabor naprav, ki jih lahko uporabimo, saj je edini pogoj, da na njej deluje Android platforma z verzijo vsaj 2.2. in kamero.

Vsak paket, se identificira s svojo številko, ki jo vpišemo v črtno kodo. Tako zmanjšamo možnost človeške napake pri vnašanju kod, saj se enostavno preberejo iz same črtne kode, ki je na paketu.



## Poglavje 2

# Primerjava konkurenčnih aplikacij

Na spletu dobimo že veliko aplikacij za android naprave, s katerimi lahko urejamo naš inventar. V okviru naloge sem ocenil nekaj izmed brezplačnih aplikacij in naredil primerjavo med njimi, kar je razvidno v Tabeli 2.1.

Ime aplikacije	Skener kod	GPS	Varnostna kopija	Izvoz	Zunanje naprave	Sinhronizacija v realnem času	Zunanji sistem za skladiščenje	Priprava dobavnice s izdelki iz skladišča
Warehouse-Assistant	Ne	Ne	Ne	Ne	Ne	Ne	Ne	Ne
Rapid Inventory	Da	Ne	Ne	Da	Da	Ne	Ne	Ne
Quick Inventory	Da	Ne	Da	Ne	Da	Ne	Ne	Ne
Inventory Counter	Da	Da	Ne	Da	Ne	Ne	Ne	Ne
My Inventory	Da	Ne	Da	Da	Ne	Ne	Ne	Ne
Warehouse Manager	Da	Ne	Ne	Da	Ne	Ne	Ne	Ne
Control Inventory	Da	Ne	Ne	Da	Ne	Ne	Ne	Ne
Inventory Manager	Ne	Ne	Da	Ne	Ne	Ne	Ne	Ne
<b>Naša aplikacija</b>	<b>Da</b>	<b>Ne</b>	<b>Da</b>	<b>Da</b>	<b>Ne</b>	<b>Da</b>	<b>Da</b>	<b>Da</b>

Tabela 2.1: Primerjava aplikacij po opcijah, ki jih ponujajo.

## 2.1 Pregled drugih aplikacij

### 2.1.1 Warehouse-Assistant

Warehouse Asistant[8] nam omogoča vnašanje artiklov in njihovih podatkov v lokalno shranjeno podatkovno bazo. V aplikacijo lahko črtne kode vnašamo samo ročno in ta ne podpira priklopa zunanjšega bralnika črtnih kod.

Vmesnik vsebuje iskalnik, preko katerega hitro najdemo želeni artikel in njegove podatke. Prednost aplikacije je, da nam omogoča pripravo naročil za vnesene artikle. Ponudi nam možnost vnosa zelene količine naročila, iz katere nato generira excel dokument. Slabost aplikacije je v tem, da se dokument shrani na napravo in ga moramo pridobiti s pomočjo datotečnega brskalnika.

Prav tako ne omogoča varnostnega kopiranja in izvoza podatkov.

### 2.1.2 Rapid Inventory

Gre za aplikacijo[9], ki nam omogoča vnašanje artiklov preko vmesnika za skeniranje kod ali ročno. Podatki se shranjujejo lokalno na telefon, omogoča pa nam izvoz v XML, CSV ali pa Google Drive Spreadsheet.

Vmesnik ima filter, s katerim lahko hitro iščemo med izdelki, ki jih imamo na razpolago ter sortiranje po šifrantih. Zanimiva opcija aplikacije je, da nam ponuja možnost priklopa zunanjšega bluetooth scannerja.

### 2.1.3 Quick Inventory

Je aplikacija[10], ki nam omogoča shranjevanje artiklov v skupine ali grupe. Vsak artikel, ki ga vnesemo, mora imeti svojo skupino, v katero se shrani.

Omogoča nam tudi izris grafa, iz katerega je razvidna količina artiklov po skupinah. Vsebuje tudi več načinov izvoza varnostne kopije, ki jih lahko shrani na SD kartico, Dropbox ali pa pošiljanje na email naslov. Vsi izmed naštetih izvozov so varnostne kopije tabele in jih ne moremo odpreti na drugih napravah.

Plačljiva verzija omogoča tudi priklop zunanjega skenerja preko USB vrač (če jih android naprava ima) ali bluetooth povezave.

### 2.1.4 Inventory Counter

Omogoča nam vnašanje kod preko kamere tudi za opis artikla in lokacije[11]. Za lokacijo obstaja možnost vnosa GPS lokacije, ki si zapomni najbližji veljavni naslov. Prednost pri vnašanju artiklov s to aplikacijo je, da vsebuje možnost dodajanja slike h artiklu. Pri vnašanju nam ponuja tudi hiter števec, ki se s klikom povečuje in kalkulator.

Filter za iskanje artiklov ponuja možnost ročnega vnosa in skeniranje črtnih kod, preko katere poišče pravi artikel.

Za izvod imamo na voljo CSV datoteko, ki jo lahko pošljemo na email naslov. Izvožena datoteka vsebuje samo osnovne podatke.

### 2.1.5 My Inventory

To je aplikacija[12], ki nam ponuja možnost vnosa artiklov ročno ali preko skeniranja kod. Pri vsakem artiklu imamo možnost dodajanja slike. Če vnašamo artikle, ki so si med seboj podobni, ima pri vsakem vnosnem polju možnost zgodovine, ki nam ponudi zadnje vnose.

Vsebuje tudi možnost iskanja po črtni kodi in imenih šifrantov. Dodatek aplikacije je, da lahko po datumu sledimo kdaj so bili artikli vneseni v aplikacijo in kdaj izpisani. Ponuja tudi enostaven graf, ki nam po dnevih izpiše število vnosov in izpisov.

Ponuja nam možnost izvoza podatkov v XML ali CSV, ter varnostno kopijo, ki se shrani lokalno na napravo.

### 2.1.6 Warehouse Manager

Je aplikacija[13], ki nam omogoča sledenje večjemu številu skladiščnih prostorov. Omogoča nam vnos večjega števila tabel, ki imajo ločene elemente.

Po skeniranju črtne kode, aplikacija preveri, če že ima artikel s kodo, ter nam ga v tem primeru izpiše. V nasprotnem primeru se nam odpre vnosna forma, preko katere vpišemo manjkajoče podatke ter novi artikel shranimo.

Dodatna funkcionalnost aplikacije je, da ji lahko nastavimo spodnjo mejo števila artiklov, ki jih želimo imeti na razpolago. Če pademo pod to mejo, se artikel obarva rdeče in nas s tem opozori.

Aplikacija ne podpira izvoza podatkov in varnostnega kopiranja.

### 2.1.7 Control Inventory

Control Inventory[14] nam omogoča vnašanje artiklov na več različnih seznamov. Aplikacija ima enostaven vmesnik, preko katerega lahko dodamo le osnovne parametre ter skeniramo črtno kodo. Prikaz seznama artiklov je mogoče sortirati po številu elementov in imenu artikla.

Aplikacija omogoča izvoz seznamov v CSV ali XML obliki. Dodatek, ki ga aplikacija ponuja je, da lahko generirani izvoz posredujemo naprej preko vseh medijev za objavo virov. S tem se poenostavi prenos podatkov na druge naprave, kjer lahko datoteko pregledujemo in urejamo.

### 2.1.8 Inventory Manager

Je razširjena aplikacija[15], ki nam omogoča sledenje artiklom, ki jih imamo na razpolago. Obrazec za vnašanje in iskanje artiklov nam ne omogoča vnosa ali skeniranja črtne kode, omogoča nam pa vnos prodajne in nakupne cene artikla.

Pri pregledu artiklov imamo možnost označitve koliko smo jih prodali, aplikacija pa po prej vneseni prodajni ceni izračuna končni dobiček pri artiklu.

Ponuja nam tudi možnost izrisa grafa, kjer lahko vidimo, kateri izmed artiklov se je najboljše prodajal, in s katerim smo imeli največji zaslužek.

Nimamo možnosti izvoza podatkov, obstaja pa opcija za varnostno kopiranje, ki podatke pošlje na email naslov.

## 2.2 Povzetek pregleda

Iz analize aplikacij, ki jih trenutno imamo na razpolago za android naprave, lahko vidimo, da se zelo razlikujejo med seboj. Vsaka izmed njih je specifična, narejena za eno vrsto skladiščenja. Nekatere izmed aplikacij so razvite bolj v smer sledenja paketom in podpirajo GPS ter sliko artikla, druge so bolj izpopolnjene za pripravo naročil. Nobena izmed njih pa ne podpira skupnega podatkovnega skladišča.

Če primerjamo našo aplikacijo z vsemi naštetimi, je največja pridobitev skupna baza in uporaba na večjem številu naprav istočasno ter urejanje in izvajanje dobavnic.

Kot večina aplikacij tudi naša uporablja bralnik kod za večjo natančnost pri vnosu podatkov. Uporabljamo tudi sistem za sledenje paketom na lokacijah, kar večina od naštetih aplikacij nima. Sistem lahko sam logično razporeja pakete po določenih parametrih in tako zagotavlja večjo učinkovitost pri skladiščenju. Dodatek, ki bi ga lahko tukaj še dodali, bi bilo GPS sledenje in bi bil učinkovit le v primeru, da je skladiščni prostor dobro signalno pokrit.

Seveda imamo na razpolago za urejanje podatkov tudi različico v spletnem brskalniku, ki nam olajša delo, saj se podatki veliko hitreje vnašajo, kot bi se preko mobilne naprave.





## Poglavje 3

# Uporabljene tehnologije

Pri razvoju aplikacije smo uporabili različne tehnologije in aplikacije. V nadaljevanju bom predstavil katere tehnologije so bile uporabljene in kakšen je njihov namen.

### 3.1 Android

Android je odprtokodni programski jezik in operacijski sistem za pametne telefone ter ostale prenosne naprave. Eden izmed popularnih platform za prenosne naprave se je hitro razširil, saj se lahko prilagodi in nima velikih sistemskih zahtev. Ker je Android odprtokoden, omogoča cenejše in lažje razvijanje programov[2].

Naša aplikacija je kompatibilna z vsemi napravami, ki uporabljajo android sistem verzije 2.2 (Froyo) ali novejši.

### 3.2 REST

Je dogovorjen slog arhitekture pošiljanja zahtev in prejemanja odgovorov. V klic, ki ga sestavlja pot do aplikacije, dodamo parametre, ki so potrebni za pravilno delovanje. V odgovor dobimo podatke, ki so bili vrnjeni s strani aplikacije.

Vrnjeni podatki so lahko enostavni teksti ali pa se uporabljajo naprednejše načine formatiranja – kot v našem primeru – JSON, XML, ipd.

### 3.3 BarcodeScanner

Za skeniranje črtnih kod smo uporabili knjižnico ZXing[1], ki vsebuje aplikacijo Barcode Scanner za Android. ZXing knjižnico smo uporabili zato, ker je odprtokodna in enostavna za integracijo. Podpira tudi več različnih vrst kod, vendar smo v našem primeru večinoma uporabljali EAN črtno kodo.

Knjižnica omogoča branje podatkov preko kamere na napravi, ki jih nato posreduje aplikaciji, iz katere je prišel klic. Aplikacija je integrirana v knjižnico, zato se namestitev izvede avtomatsko.

ZXing je ena izmed najbolj razširjenih in najnovejših knjižnic za branje črtnih kod. Konkurenčna je tudi knjižnica ZBar[17], vendar že nekaj časa ne podpira zadnjih verzij Androida.

## Poglavje 4

# Analiza aplikacije

### 4.1 Namen in uporaba aplikacije

Za boljši nadzor nad sredstvi, je potrebno organizirano skladišče, kot vidimo primer na Sliki 4.1. Za to pa je potreben dober in organiziran sistem.

Aplikacija je v osnovi namenjena za uporabo v skladiščnih sistemih. Njen namen je olajšati delo uporabnikom, saj ima uporabniku prijazen vmesnik (na android platformi), preko katerega lahko kadarkoli dostopajo do podatkov in funkcionalnosti, ki jih pri svojem delu potrebujejo.

Seveda je včasih potrebno izvesti tudi funkcije, ki jih ta del aplikacije nima (npr.: brisanje podatkov, ročno premikanje, ...), saj hočemo zmanjšati možnost napake in izgube podatkov.

Za ta namen imamo na drugi strani tudi naprednejši del aplikacije, ki nam omogoča zahtevnejše in nevsakdanje operacije. Do njega imajo dostop samo nadzorni operaterji in administratorji sistema, saj preko nje dostopamo neposredno do podatkovnega skladišča. Tukaj imamo nabor zahtevnejših operacij, do katerih uporabniki na prenosnih napravah ne morejo dostopati, saj je za dostop potrebna avtentikacija uporabnika z uporabniškim imenom in geslom.

Aplikacija je sestavljena tako, da se lahko prilagodi za različne namene uporabe, edini pogoj je da se pri tem uporablja črna koda, zato je primerna

za velika ter manjša skladišča.

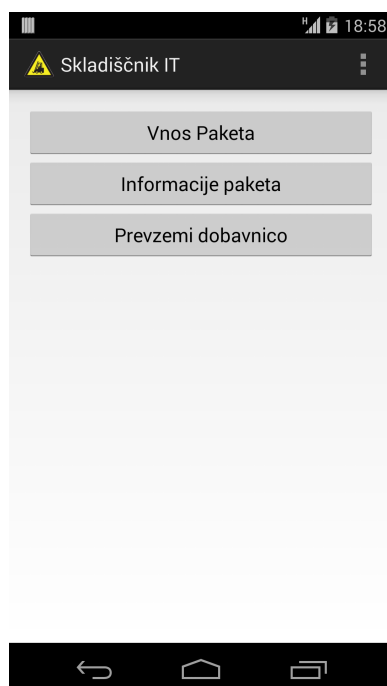


Slika 4.1: Primer dobro organiziranega skladišča.

## 4.2 Delovanje na prenosni napravi Android

Na napravi Android je potrebno namestiti aplikacijo, ki nam služi kot vmesnik med sistemom in uporabnikom. Zaslonski posnetek aplikacije je viden na Sliki 4.2. Aplikacijo sestavlja vmesnik, ki skrbi za prenos podatkov med strežnikom in napravo ter integrirani sistem za prebiranje črtnih kod – aplikacija Barcode Scanner [1]. Za prenos podatkov, ki jih potrebujemo za delovanje, se aplikacijo poveže v mrežo preko brezžične povezave in dostopa do strežnika preko IP naslova.

Delo na aplikaciji poteka tako, da uporabnik izbere operacijo, ki jo želi izvesti preko menija ter sledi navodilom, ki mu jih aplikacija postavi. Ker je ta del sistema odprt in do njega lahko dostopa vsak, ima uporabnik omejeno funkcionalnost. Aplikacija ponuja možnost skladiščenja novega paketa, preverjanje informacij paketa in prevzem dobavnice, ki združuje večje število paketov v celoto [3].



Slika 4.2: Zaslonska maska aplikacije na Android napravi.

## 4.3 Delovanje na strežniku - PHP in MySQL

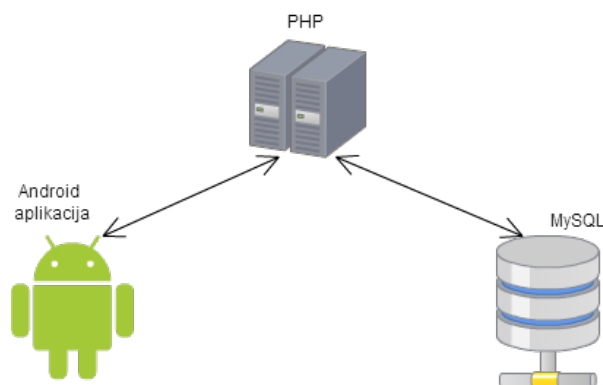
Strežniški del aplikacije je sestavljen iz dveh delov:

- podatkovnega skladišča MySQL,
- aplikacije za nadzor in urejanja podatkov v PHPju.

Aplikacija se deli še na dva sistema – administrativni sistem in sistem za komunikacijo s Android napravo. Administrativni sistem je namenjen naprednim uporabnikom, saj ponuja več funkcionalnosti, kot jo imamo na Android napravi, med drugim tudi možnost brisanja. Ker je dostop do podatkov bolj direkten, je sistem zaščiten s uporabniškim imenom in geslom, da preprečimo nezaželene dostope.

Komunikacijski sistem skrbi za povezavo med Android aplikacijo in podatkovnim skladiščem, povezava je razvidna na Sliki 4.3. Ko se izvede določena zahteva v aplikaciji, pošlje poizvedbo do sistema, ki nato preveri in zbere

želeno podatke, ter vrne odgovor. Sistem lahko odgovori tudi s kodo napake, ki jo nato aplikacija predstavi uporabniku. Na ta način ločimo interni ter zunanji sistem ter bolj zaščitimo podatkovno skladišče pred zunanjimi vplivi ter zmanjšamo možnost napak in vdorov v sistem.



Slika 4.3: Povezava Android aplikacije s podatkovnim skladiščem.

## 4.4 Prednost pred ostalimi sistemi

Prednost pred ostalimi sistemi za skladiščenje je, da se lahko hitro prilagodi in ne potrebuje točno določene strojne opreme. Aplikacijo za skladiščenje lahko zaženemo na praktično vsaki Android napravi, za delovanje ne potrebujemo fizičnih skenerjev. Strežniški del je zgrajen tako, da je obremenitev sistema minimalna, saj se transakcije izvajajo samo, ko je bila zanje prejeta zahteva.

Aplikacije za skladiščenje, ki jih trenutno dobimo za Android naprave, imajo to pomanjkljivost, da shranjujejo podatke v datoteke, ki se nahajajo fizično na telefonu. To pomeni, da bi v primeru večjega števila uporabnikov, vsak hranil podatke samo zase in ne bi imel vpogleda v transakcije, ki jih je izvedel nekdo pred njim. Naša aplikacija hrani podatke v podatkovnem skladišču, do katerega imajo vsi uporabniki dostop v realnem času.

# Poglavje 5

## Arhitektura sistema

Predstavitev arhitekture sistema je sestavljena skladno s Kruchtenovim[16] arhitekturnim modelom 4+1.

Pri tem sem uporabil pogled primerjave uporabe, implementacijski pogled, ki je združen s fizičnim pogledom, procesni in podatkovni pogled.

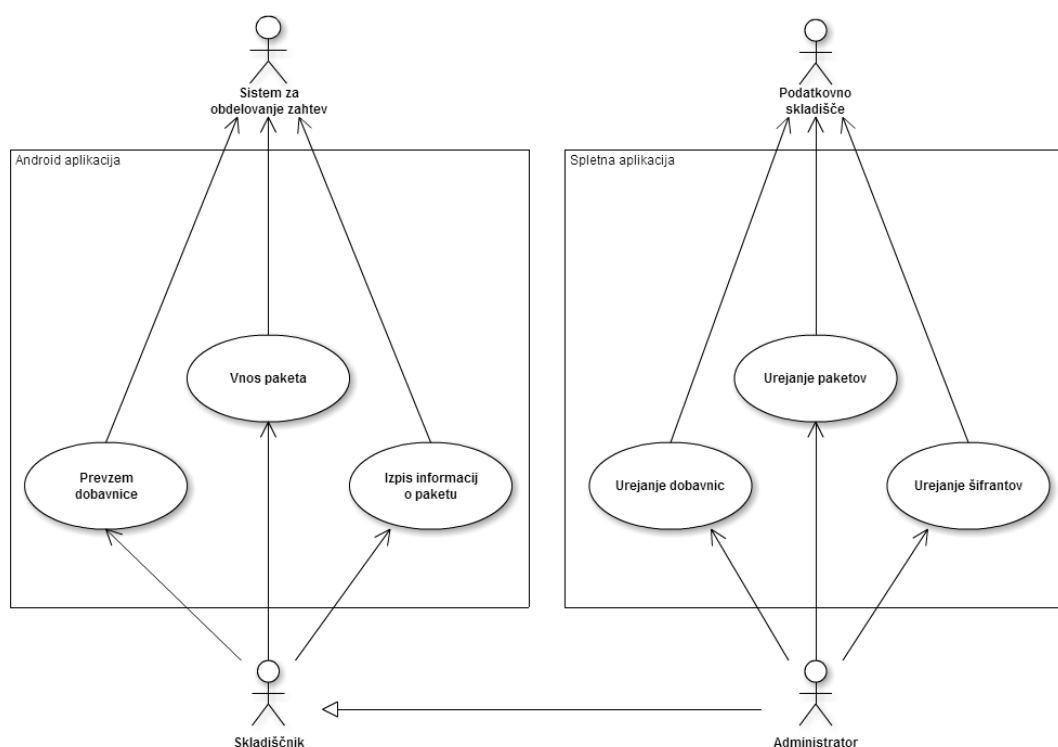
### 5.1 Pogled na primere uporabe

Kot je razvidno iz Slike 4.1, sta glavna uporabnika celotnega sistema skladiščnik in administrator, ki ima vse pravice skladiščnika.

Skladiščnik lahko dostopa preko android naprave do aplikacije in pri tem uporablja skladiščni sistem. Sistem je razdeljen na več manjših procesov aplikacije, ki izvajajo različne akcije. Skladiščnik lahko vnese paket, izvede poizvedbo o informaciji paketa ali pa prevzame dobavnico.

Administrator ima dodatno dostop do administrativnega sistema, do katerega dostopa preko web vmesnika. Tukaj lahko dodaja in ureja šifrante ter upravlja pakete in sestavlja dobavnice.

Oba sistema sta povezana s podatkovnim skladiščem, ki predstavlja MySQL bazo podatkov. Glede na aplikacijo, preko katere urejamo podatke, lahko izvajamo določene operacije, s katerimi dodajamo, urejamo in brišemo te podatke.



Slika 5.1: Diagram primerov uporabe.

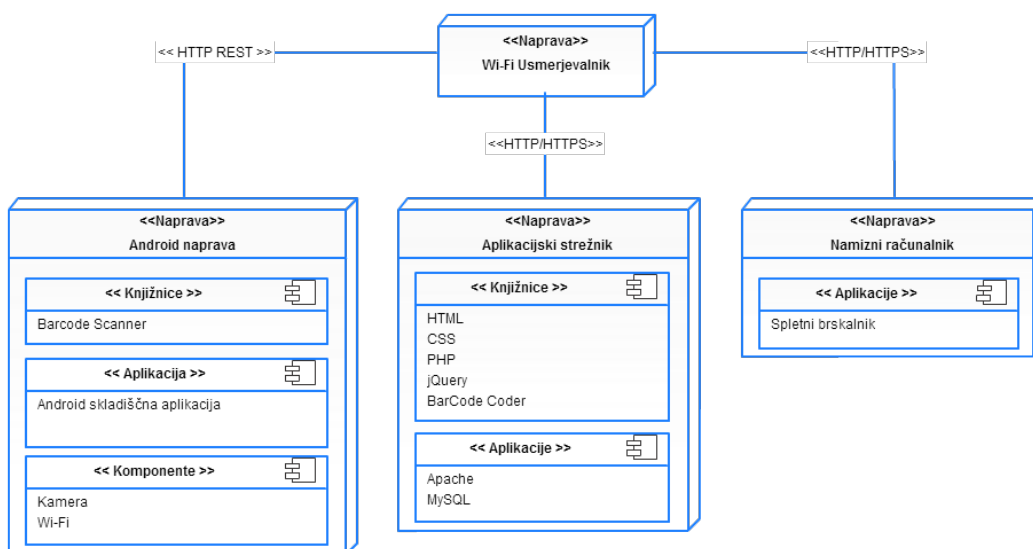
## 5.2 Implementacijski pogled

Na Sliki 4.3 je viden komponentni diagram, ki vsebuje vse dele, ki jih sistem potrebuje za delovanje. Za zbiranje podatkov mora android naprava vsebovati kamero. Za branje podatkov iz kamere uporablja aplikacijo Barcode scanner, ki je del odprtokodne knjižnice ZXing [1]. Prenos podatkov, ki so se zbrali ob skeniranju, se posreduje preko klicev oz. intentov. Naprava android komunicira s aplikaciskim strežnikom, preko Wi-Fi usmerjevalnika za prenos podatkov pa uporablja HTTP REST protokol.

Usmerjevalnik je povezan s aplikaciskim strežnikom, ki procesira vse dobljene zahteve in obdeluje podatke v podatkovnem skladišču.

Za administracijski dostop je potrebna še prijava preko spletnega brskalnika, za katero pa potrebujemo osebni računalnik.





Slika 5.2: Komponentni diagram sistema v povezavi z vozlišči.

## 5.3 Procesni pogled

V Sliki 4.2, opisujemo aktivnosti, ki so potrebne za vskladiščenje paketa. Pri izvedbi procesa imamo tri ključne elemente. Skladiščnik je fizična oseba, ki uporablja sistem in sledi navodilom, ki so mu posredovana. Android aplikacija je vmesnik med končnim uporabnikom in centralno aplikacijo, skrbi za prenos in interpretacijo podatkov med obema. Centralna aplikacija je sistem, ki glede na prejete podatke izvede določene funkcije in posreduje dobljen odgovor.

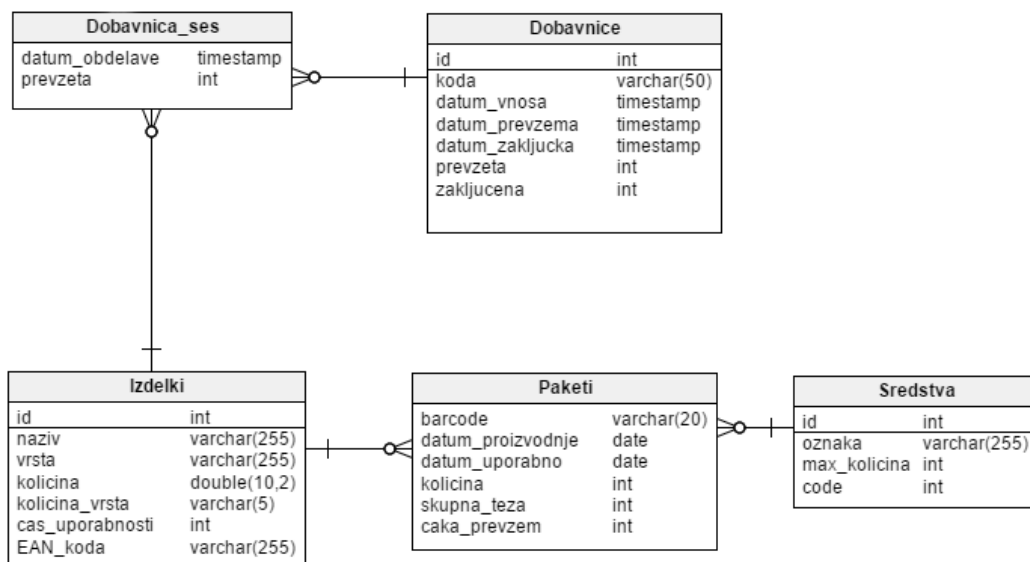
Celoten proces se začne s zaključkom paketa in skeniranjem etikete. Če so podatki o paketu najdeni v bazi, jih aplikacija prikaže, v nasprotnem primeru se vrnemo na korak skeniranja etikete.

Po potrditvi pravilnosti podatkov, skladiščnik sproži naslednji korak, kjer se začne skladiščenje. Aplikacija nam v primeru prostega mesta ponudi njegovo lokacijo, drugače nas aplikacija opozori, da prostega mesta v skladišču ni in zaključi proces.

Skladiščnik ob prihodu skenira kodo na lokaciji. Če je koda pravilna, nas



izdelki, ki jih vsebuje dobavnica. Uporabljamo jo za sledenje izdelkom in preverjamo, kateri paketi so že bili prevzeti.



Slika 5.4: ER diagram podatkovne baze.



# Poglavje 6

## Delovanje sistema

### 6.1 Delovanje Android aplikacije

Vmesnik, ki povezuje skladiščnika in sistem, je v osnovi razdeljen na 3 osnovne funkcije, ki jih lahko izvaja:

- vnos paketa,
- poizvedbe o informaciji paketa in
- prevzem dobavnice.

Naprava Android je preko brezžičnega omrežja povezana do centralnega sistema[5]. V ozadju sistem sam izvaja potrebne procese in beleži podatke. Aplikacija ob vsaki poizvedbi ali vnosu pošlje paket na sistem, kjer se podatki obdelajo in posredujejo odgovor.

Pri predstavitvi delovanja smo aplikacijo uporabljali v tovarni, kjer proizvod predstavljajo paketi, ki jih je potrebno skladiščiti. Vsak paket po izdelavi dobi svojo črtno kodo, v sistem pa se zapišejo vsi potrebni podatki. Ko aplikacija prevzame paket za skladiščenje, sistem preveri razpoložljiva mesta in paket, glede na vrsto in datum proizvodnje, shrani na ustrezno mesto. Vse lokacije so označene s črtnimi kodami. Skladiščnik prejme podatke o lokaciji, na katero shrani paket.

Po tem, ko paket odloži, mora zaključiti svoj proces s skeniranjem kode na lokaciji, in s tem zaključi postopek vnosa paketa.

### 6.1.1 Vnos paketa

Ko pride paket na konec proizvodnje linije, se nanj prilepi označba s črtno kodo, ki jo pripravi sistem. V aplikacijo se vpišejo vsi podatki, ki so potrebni za identifikacijo paketa, nato pa se zgenerira črtna koda. Eden izmed pomembnih delov dobro zasnovanega sistema za skladiščenje je logika shranjevanja. Sistem mora biti pri shranjevanju efektiven in fleksibilen, saj nam mora omogočati možnost dodajanja novega skladiščnega prostora.

V našem sistemu uporabljamo več nivojske prehodne regale. Omogočajo nam shranjevanje v več nadstropij in možnost vnašanja in pobiranja iz obeh strani. Sistem deluje tako, da se vedno nove pakete odlaga v smeri označeni na regalu. Tako imamo na drugi strani, kjer se pakete pobira, vedno artikole s starejšim datumom, ki jih moramo odpremiti najprej.

Ker je aplikacija, ki skrbi za skladiščenje na strežniku, se istočasno lahko izvaja več vnosov paketov. Ko aplikacija prejme zahtevek za vnos, rezervira prostor, na katerega se bo skladiščil paket, tako v času transporta ne more drugi paket zasesti določenega prostora [4].

### 6.1.2 Postopek skladiščenja

Takoj, ko paket pride iz proizvodnje linije in mu je dodeljena črtna koda, lahko začnemo s skladiščenjem. Ob prevzemu skladiščnik skenira kodo na paketu, aplikacija pa mu preveri stanje.

V primeru, da je bil paket že prevzet in je bil samo odložen, aplikacija vrne opozorilo, da je vnos paketa v teku. Skladiščnik tega paketa ne more prepeljati na določeno lokacijo, to lahko zaključi le uporabnik, ki je s to operacijo začel.

Če je paket že vskladiščen, ko ga uporabnik skenira, mu aplikacija vrne opozorilo, da je paket že bil vskladiščen. Uporabnik nima opcije za skladiščenje,

lahko pa preveri lokacijo paketa in ga prestavi na pravo mesto.

Ko je paket pravilno vnesen v sistem in še ni bil prevzet, mu aplikacija izpiše podrobnejše podatke o paketu. Ko skladiščnik preveri, da so podatki pravilni, izbere opcijo skladiščenja, aplikacija pa ga usmeri na lokacije, kjer se paket shrani. Primer postopka na Sliki 6.1.

V tem koraku se mu izpišeta ime lokacije in številka lokacije, ki je enaka številki črtne kode na regalu. V tem zadnjem koraku skladiščnik shrani paket na določeno lokacijo in za potrditev poskenira črtno kodo, ki je na regalu.

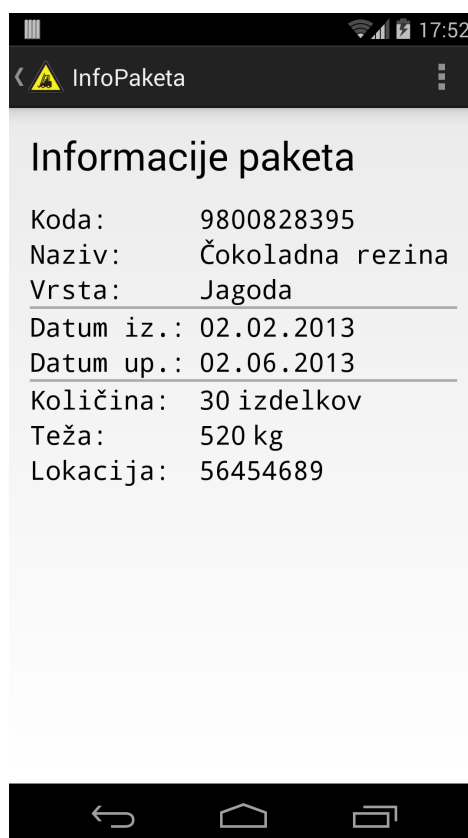
S tem je postopek zaključen in uporabnik lahko nadaljuje postopek z naslednjim paketom v vrsti.



Slika 6.1: Primer postopka skladiščenja.

### 6.1.3 Poizvedba o informaciji paketa

Če pridemo do paketa, ki je že bil vskladiščen (opcija za skladiščenje nam vrača opozorilo, da je paket že v sistemu), lahko izvedemo poizvedbo o informaciji paketa. Ta operacija nam vrne vse podatke o tem paketu in njegovo skladiščno lokacijo. Primer izpisa lahko vidimo na Sliki 6.2.



Slika 6.2: Primer prikaza podatkov po poizvedbi.

### 6.1.4 Prevzem dobavnice

Ko administrator v sistem vnese novo dobavnico in ji doda zelene pakete, jo lahko skladiščnik prevzame.

Za izvedbo prevzema, mora skladiščnik prejeti kodo dobavnice, ki jo ročno

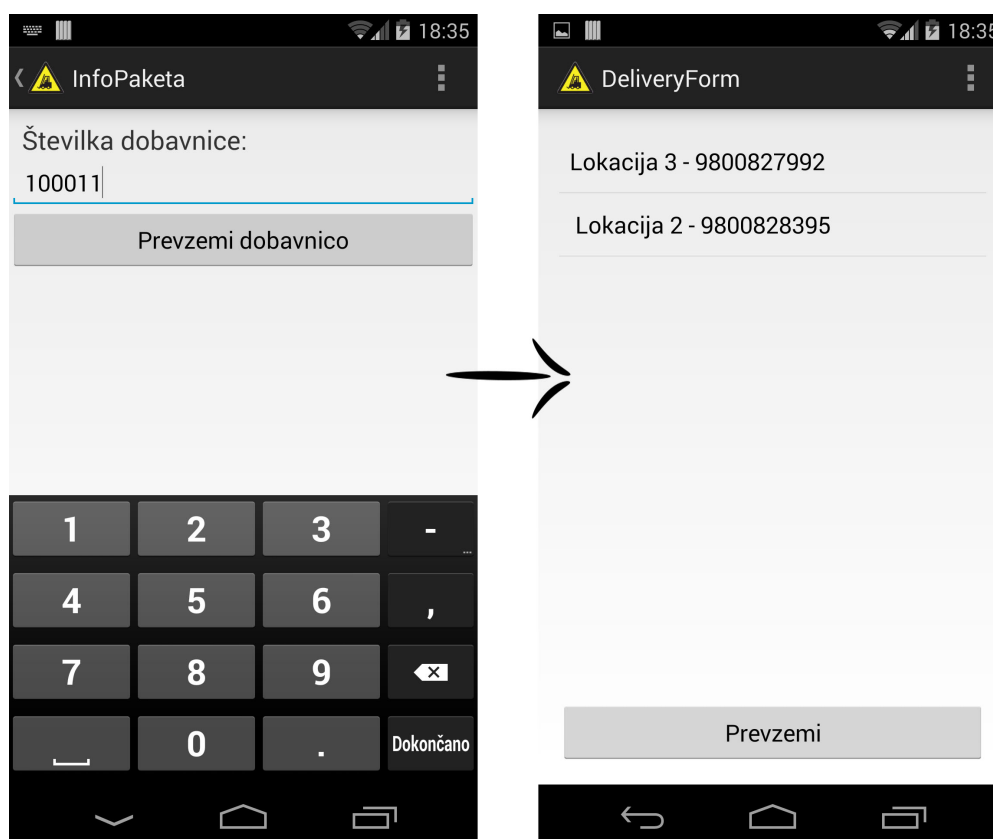


vnese v polje za prevzem. Če v skladišču nimamo dovolj sredstev za sestavo dobavnice, nas aplikacija na to opozori, prevzem dobavnice se ne izvede.

Ko je v skladišču dovolj sredstev za uspešen prevzem in skladiščnik izvede operacijo, se dobavnica označi da je v delu. S tem onemogočimo, da bi kdo drug v tem času prevzemal pakete, ki so potrebni za izvedbo.

Ob uspešnem prevzemu dobavnice aplikacija dodeli skladiščniku vsa potrebna sredstva in jih onemogoči. Tako preprečimo drugim uporabnikom, da bi v sklopu druge dobavnice prevzeli paket, ki ga mi potrebujemo za zaključitev naše.

Kot je razvidno iz Slike 6.3, nam aplikacija ponudi seznam z vsemi paketi, ki so potrebni za prevzem dobavnice in njihovimi lokacijami. Sedaj skladiščnik lahko prevzame vsak paket posebej in ga pripravi za odvoz.



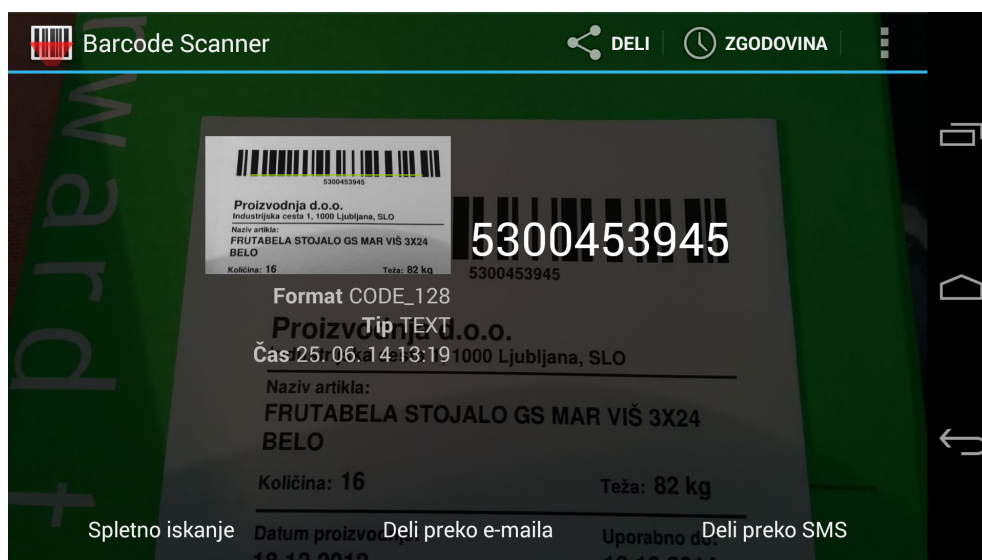
Slika 6.3: Primer postopka prevzema dobavnice.

Ko se paket prevzame, uporabnik skenira črtno kodo, ki je na njem. Če paket ustreza tistemu, ki je naveden na seznamu in mu ga je postavila aplikacija, se ta zaključi in odstrani iz seznama nalog. Tako označimo, da je bil paket pripravljen in je njegova naloga zaključena.

Ob skeniranju zadnjega paketa na seznamu, nas aplikacija opozori, da je bil prevzem dobavnice uspešno zaključen in odstrani dobavnico iz seznama nedokončanih. Postopek prevzema je zaključen, prevzem iste dobavnice ni več mogoč.

## 6.2 Uporabljene knjižnice android

Pri delovanju android aplikacije[7] smo za prenos podatkov med kamero in aplikacijo uporabili knjižnico ZXing[1], ki uporablja aplikacijo Barcode Scanner s Slike 6.4. Ta odprtokodna knjižnica nam omogoča branje črtnih kod, ki jih preberemo preko kamere ter posredovanje podatkov nazaj v aplikacijo.



Slika 6.4: Vmesnik ZXing za branje črtnih kod.

Za prenos podatkov med aplikacijami se uporablja funkcija `Intent`. Slika 6.5 nam predstavlja kako lahko s klicem funkcije zaženemo vmesnik za branje

črtnih kod, ki je aktiven, dokler ni bila skenirana veljavna koda ali do preklica uporabnika.

```
IntentIntegrator integrator = new IntentIntegrator(this);  
integrator.initiateScan();
```

Slika 6.5: Inicializacija funkcije za zagon kamere.

Po zaključku uspešnega skeniranja kode, v aplikaciji prestrežemo posredovane podatke. Primer te funkcije je razviden na Sliki 6.6. Funkcija nam vrne tri parametre `requestCode`, `resultCode` in `intent`.

`requestCode` nam vrne številčno vrednost aktivnosti, ki jo je zagnala. S tem parametrom lahko preverjamo katera aktivnost pričakuje podatke, če uporabljamo več `intent`-ov v enem procesu.

`resultCode` je parameter, s katerim lahko preverimo, kako se je vmesnik zaprl. Tako lahko ugotovimo, ali je bil zaprt s pravilnim skeniranjem kode ali je prišlo do napake oziroma nepravilnega zaključka.

`intent` je funkcijska vrednost klica aplikacije.

```
public void onActivityResult(int requestCode, int resultCode,  
    Intent intent) {  
    if (resultCode == Activity.RESULT_OK) {  
        IntentResult scanResult =  
            IntentIntegrator.parseActivityResult(requestCode,  
                resultCode, intent);  
        code = scanResult.getContents();  
        ...  
    }  
}
```

Slika 6.6: Prejemanje podatkov iz aktivnosti.

## 6.3 Delovanje strežniških aplikacij

Aplikacije, ki delujejo na strežniku, so ločene od android aplikacije[6]. Do njih lahko dostopamo samo preko administracijskega sistema, preko katerega se moramo avtentificirati z uporabniškim imenom in geslom. Strežniške aplikacije so razdeljene na dva dela:

- administrativni sistem in
- sistem za komunikacijo med napravo android in strežnikom.

### 6.3.1 Administrativni sistem

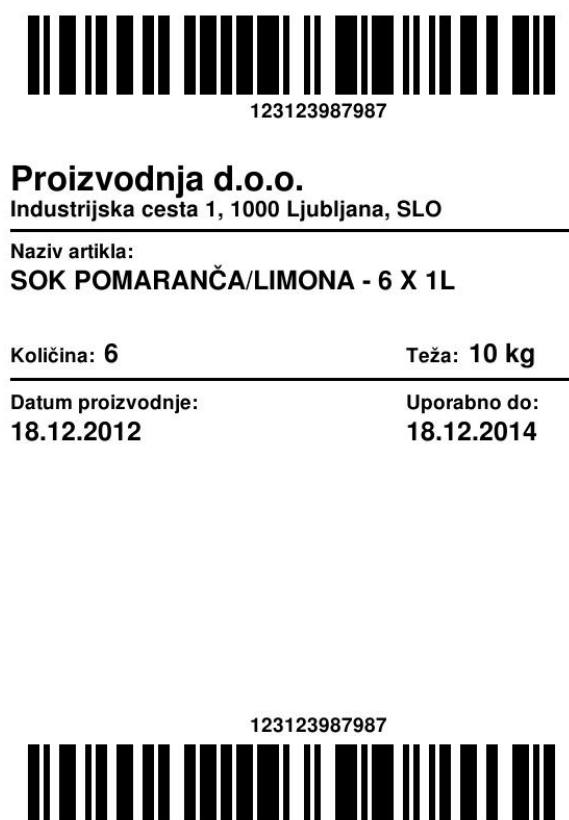
Sistem je namenjen direktnemu upravljanju s podatki, ki so shranjeni v podatkovni bazi. Do sistema je mogoče dostopati preko spletnega brskalnika. Na vhodni strani se mora uporabnik prijaviti s pravilnim uporabniškim imenom in geslom. Če so vneseni podatki napačni, sistem njegovo prijavo zavrne.

Ob uspešni avtentikaciji, nam sistem ponudi operacije za upravljanje s paketi in njihovimi šifranti:

- tiskanje etiket,
- upravljanje paketa,
- dodajanje šifrantov,
- urejanje šifrantov in
- sestavo dobavnice.

Preden lahko vskladiščimo nov paket, ga moramo označiti z etiketo. Vsakemu paketu, ki zapusti proizvodnji del, je potrebno določiti specifične podatke.

Uporabnik preko vmesnika izbere vrsto proizvoda in artikel ter vnese količino izdelka v paketu. Aplikacija vnese paket v podatkovno skladišče ter generira etiketo z EAN kodo in preostalimi podatki. Primer enostavne kode, ki jo generira program, je viden na Sliki 6.7.



Slika 6.7: Pravilno pripravljena etiketa za označevanje paketa.

Uporabnik natisne dobljeno etiketo ter jo postavi na vidno mesto paketa. Sledi skladišniški del, pri katerem potrebujemo android napravo.

Upravljanje paketa nam omogoča dodajanje in spreminjanje obstoječih podatkov paketov. Pri vnosu novega paketa nam sistem ponudi podatke iz predhodno definiranih šifrantov. Primer zaslonske maske administracijskega sistema je viden na Sliki 6.8.

Administrator lahko tukaj spreminja lokacijo in količino artiklov v paketu, če je bil nedokončan in ga ni mogoče skladiščiti preko običajnega postopka. Za uspešno dodajanje paketa je prej potrebno definirati šifrante. Z njimi enotno označujemo vrednosti izdelkov in lažje združujemo pakete z enakimi vrstami artiklov.

ID	Naziv	Vrsta	Barcode	Datum proiz.	Lokacija	<u>Briši</u>	<u>Uredi</u>
5	Čokoladna rezina	Malina, Brusnica	9800828401	2012-12-25	58795956	Izbriši	Uredi
22	Voda	Jabolko	9800712472	2013-10-02	58795956	Izbriši	Uredi
23	Voda	Limona	5300453945	2013-11-02	58795956	Izbriši	Uredi
6	Čokoladna rezina	Jagoda	9800828395	2013-02-02	56454689	Izbriši	Uredi
21	Čokoladna rezina	Jagoda	5600464634	2013-09-30	56454689	Izbriši	Uredi

Slika 6.8: Zaslonska maska vmesnika za urejanje podatkov paketa.

V sklopu šifrantov imamo dve skupini:

- izdelki in
- lokacije.

Izdelki vsebujejo vse podatke, ki jih hranimo o vrsti artikla. Brez predhodno definiranih šifrantov ne moremo vnesti novih paketov.

### 6.3.2 Sistem za komunikacijo z android napravo

Sistem, ki obdeluje zahteve poslane iz prenosnih naprav, deluje na istem PHP sistemu kot naš administrativni del. Za normalno delovanje morajo biti naprave pravilno konfigurirane z IP naslovom strežnika, na katerem deluje PHP sistem.

Vse zahteve se iz naprave pošiljajo preko URL klicev, ki vsebujejo IP naslov strežnika oz. ime gostitelja ("host name"), ime skripte in določene parametre, ki jih potrebujemo za pravilno delovanje.

Primer enostavnega klica, ki kot parameter pošilja kodo, je razviden na Sliki 6.9.

```
new downloadData().execute(  
    "http://" + serverIP + "/AndroidService/infoPaketa.php?code=" + code  
);
```

Slika 6.9: Klic skripte za izpis informacije paketa.

V skripti se izvedejo vse potrebne operacije, tudi spremembe nad podatki.

```
[{  
    "ID": "3",  
    "barcode": "9800828395",  
    "Datum_proizvodnje": "02.02.2013",  
    "Datum_uporabno": "02.06.2013",  
    "kolicina": "30",  
    "kolicina_vrsta": "g",  
    "naziv": "\u010cokoladna rezina",  
    "vrsta": "Jagoda",  
    "EAN_koda": "2123456561411",  
    "lokacija": "2",  
    "oznaka": "Lokacija 2",  
    "code": "56454689"  
}]
```

Slika 6.10: JSON odgovor v primeru uspešnega klica.

Skripta nam nato vrne odgovor v JSON fromatu. Primer vrnjenega JSON paketa je razviden na Sliki 6.10. V primeru napake, je odgovor prazen oz. vsebuje številko napake za lažje ukrepanje. V nasprotnem primeru, android aplikacije prejme odgovor iz PHP skripte ter uporabi prejete podatke za nadaljnje operacije.





## Poglavje 7

# Zaključek

Kljub temu, da so aplikacije za skladiščenje razširjene, se je izkazalo, da nobena od trenutno dostopnih aplikacij ne podpira uporabe večjega števila naprav naenkrat in skupnega podatkovnega skladišča. Ta problem sem rešil v okviru diplomske naloge tako, da se s pomočjo android naprave, podatke posreduje na strežnik, kjer so dostopni v realnem času.

Prikazali smo delovanje sistema, od ideje do končnega delujočega paketa aplikacij. Za samo delovanje sistema potrebujemo tudi nekaj strojne opreme, ki pa ni ravno zahtevna, zato se lahko aplikacijo uporabi tudi v manjših skladiščnih okoljih.

Odziv aplikacije v testnem okolju pri optimalnih okoliščinah je hiter, za praktično uporabo v skladišču, pa bi se znale pokazati pomanjkljivosti pri branju črtne kode. Trenutno se koda prebere iz slike, ki jo zajamemo s telefonom. V primeru, da je kvaliteta slike preslaba ali pa je premalo osvetljena, je uporaba skenerja nezanesljiva.

V primeru večjega skladiščnega prostora, bi bilo potrebno aplikacijo nadgraditi še s GPS sledenjem, tako da bi nas pri izbiri lokacije tudi usmerila v pravo smer.



# Literatura

- [1] Official ZXing ("Zebra Crossing") project home, dostopna na:  
<https://github.com/zxing/zxing>, (Oktober) 2014
- [2] Android developer - Developer Tools, dostopna na:  
<http://developer.android.com/develop/>, (Oktober) 2014
- [3] Charlie Collins, Michael D. Galpin, Matthias K  ppler *Android in practice*. Addison-Wesley, 2012.
- [4] Firtman, Maximiliano R. *Programming the mobile web*. O'Reilly, 2010.
- [5] Shane Conder, Lauren Darcey *Android wireless application development*. Addison-Wesley, 2010.
- [6] Mednieks, Zigurd R., Dornin, Laird, Meike, G. Blake, Nakamura, Masumi *Programming Android*. O'Reilly, 2011.
- [7] Luke Welling, Laura Thomson *PHP and MySQL web development*. Addison-Wesley, 2009.
- [8] igLabs - Warehouse-Assistant, Google play aplikacija, dostopna na:  
<https://play.google.com/store/apps/details?id=iglabs.android.inventory>,  
(Oktober) 2014
- [9] TEC-IT - Rapid Inventory, Google play aplikacija, dostopna na:  
<https://play.google.com/store/apps/details?id=com.tecit.inventory.android.demo>,  
(Oktober) 2014

- 
- [10] aadhk - Quick Inventory, Google play aplikacija, dostopna na:  
<https://play.google.com/store/apps/details?id=com.aadhk.inventory>,  
(Oktober) 2014
- [11] Fernando Haro - Inventory Counter, Google play aplikacija, dostopna na:  
<https://play.google.com/store/apps/details?id=es.fernandoharo.inventario>,  
(Oktober) 2014
- [12] CamelDroid - My Inventory, Google play aplikacija, dostopna na:  
<https://play.google.com/store/apps/details?id=com.appdevandroid.framework.inventory>,  
(Oktober) 2014
- [13] PigeonSoftware - Warehouse Manager, Google play aplikacija, dostopna  
na:  
<https://play.google.com/store/apps/details?id=pigeonsSoftware.WarehouseManagerFR>,  
(Oktober) 2014
- [14] CMDJ - Control Inventory, Google play aplikacija, dostopna na:  
<https://play.google.com/store/apps/details?id=com.chato.app>, (Okto-  
ber) 2014
- [15] Idempotent Systems - Inventory Manager, Google play aplikacija,  
dostopna na:  
<https://play.google.com/store/apps/details?id=com.trinisoftinc.invman>,  
(Oktober) 2014
- [16] Philippe Kruchten - Architectural Blueprints—The “4+1” View Model  
of Software Architecture, dostopna na:  
[http://plg.uwaterloo.ca/~holt/cs/446/08/slides/4+1view-  
architecture.pdf](http://plg.uwaterloo.ca/~holt/cs/446/08/slides/4+1view-architecture.pdf), (Oktober) 2014
- [17] ZBar bar code reader, dostopna na:  
<http://zbar.sourceforge.net/>, (Oktober) 2014